LeanIX

Kickstart a Design System Revolution with Storybook

Konstantin Tieber

Angular-Cologne Meetup, November 13th 2019

xkons.de

What is Storybook?

LeanIX

- Development environment for UI components
- Interactive catalogue for UI components
- Supports all relevant frameworks and more
- Comes with lots of handy addons
  - Knobs, Accessibility, Actions, Backgrounds, Viewport, Themes...

Addons Demo

LeanIX

Who needs Storybook?

Storybook Setup

LeanIX

- Framework specific guides
  https://storybook.js.org/docs/guides/guide-angular/

- Setup: npx -p @storybook/cli sb init --type angular

- Start: npm run storybook

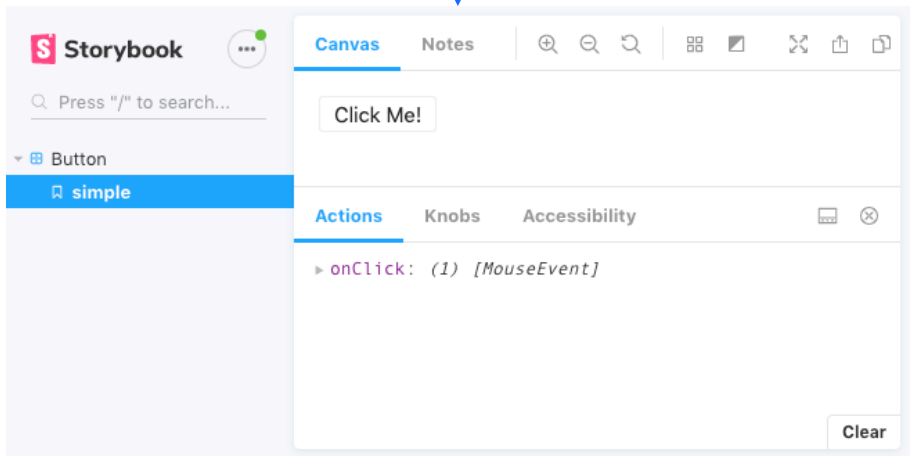
Writing Stories

LeanIX

```
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { moduleMetadata, storiesOf } from '@storybook/angular';
import { ButtonComponent } from 'lx-core-ui/components';
import { action } from '@storybook/addon-actions';

storiesOf('Button', module)
  .addDecorator(
    moduleMetadata({
      imports: [BrowserAnimationsModule]
    })
  )
  .add('simple', () => ({
    component: ButtonComponent,
    template: `<button [label]="label" (click)="onClick()"></button>`,
    props: {
      label: 'Click Me!',
      onClick: action('onClick')
    }
  }));
```

```
.
├── src
│   └── components
│       └── button.js
└── stories
    └── button.stories.js
```



Storybook 5.2 🥳

LeanIX

Old „storiesOf“ style

```
import { storiesOf } from '@storybook/react';

storiesOf('atoms/Button', module)
  .add('text', () => <Button>Hello</Button>)
  .add('emoji', () => <Button>😀😎👍💯</Button>);
```

New Component Story Format (CSF)

```
export default { title: 'atoms/Button' };

export const text = () => <Button>Hello</Button>;
export const emoji = () => <Button>😀😎👍💯</Button>;
```

Works in any environment that supports ES6 JavaScript!

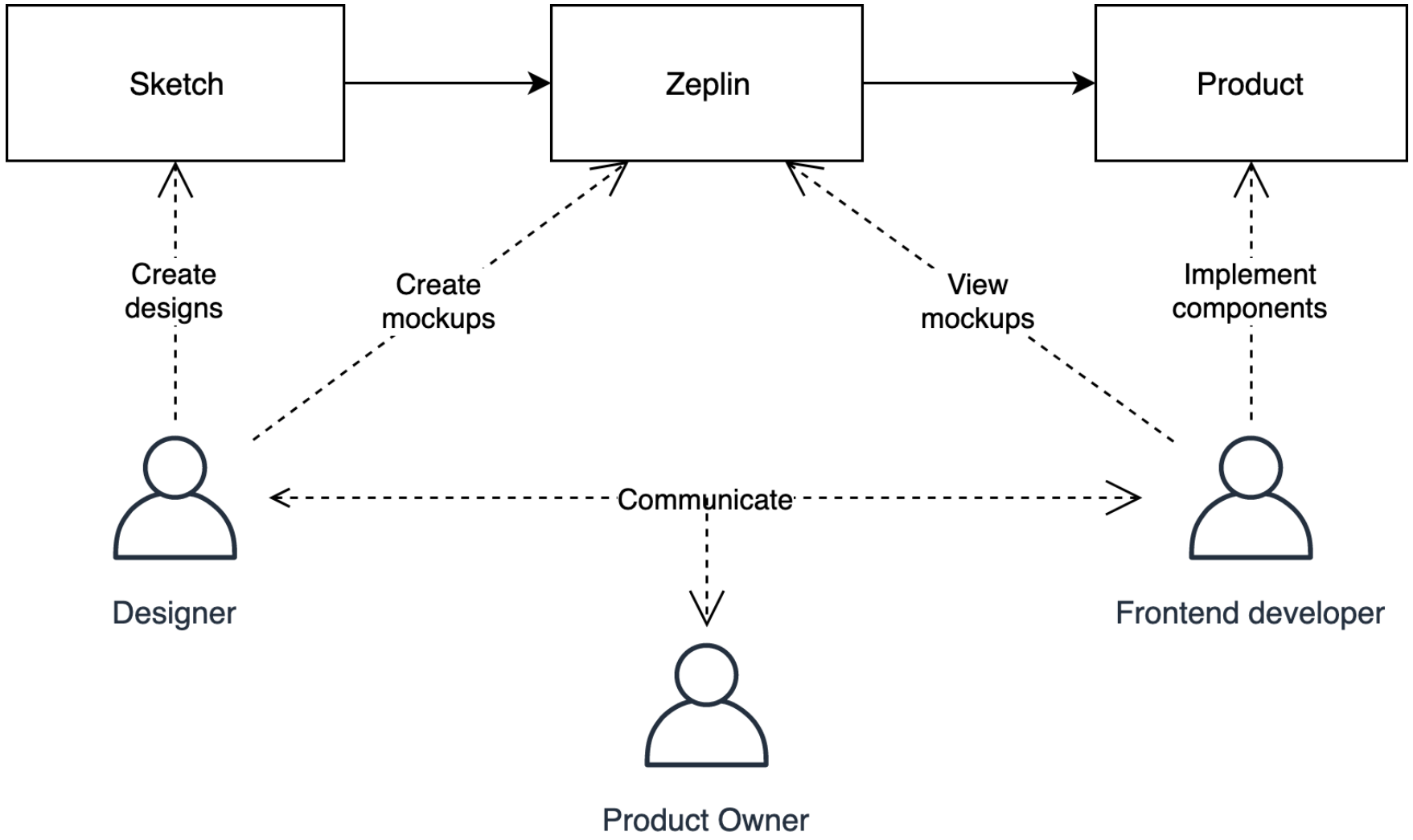
Quelle: https://medium.com/storybookjs/storybook-5-2-794958b9b111/

LeanIX

Storybook at LeanIX

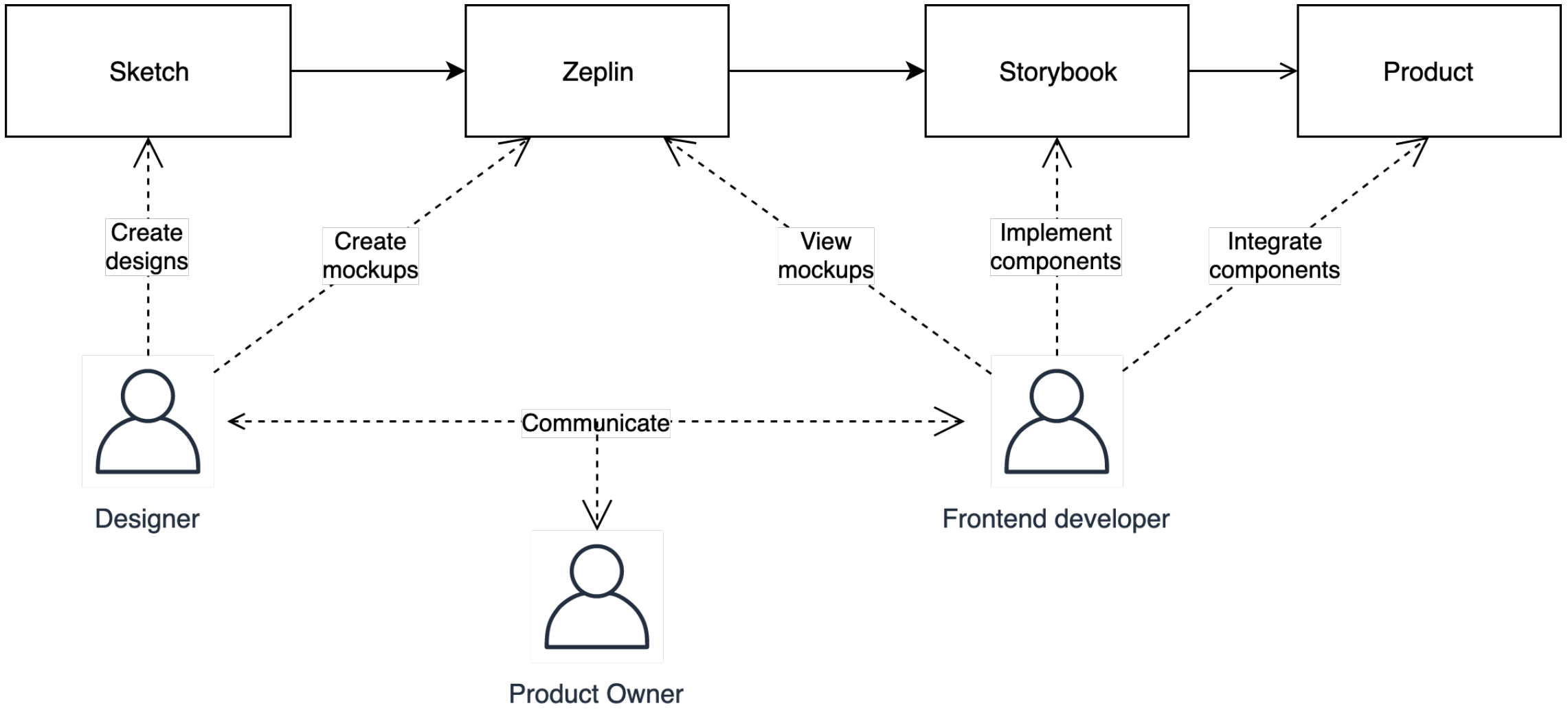
Design Process without Storybook

LeanIX



Design Process with Storybook

LeanIX



Benefits for Designers

- Test the real components before they've been integrated into the product.
- Catalogue of existing components, their states and usability.
- Convenient playground for discussions and acceptance.

Benefits for Developers

LeanIX

- Faster rerendering thanks to smallest possible app to render component and Hot Module Replacement
- Makes you think twice about how to structure your Angular modules.
- Encourages writing “dumb“ components.

Dumb / Presentational components

LeanIX

- are only concerned with the presentation of a component.
  - no API requests
  - don‘t know the Store (“in memory database“ with application state)
  - no state
    - functional components in React
    - onPush changeDetection in Angular (props will only change from the parent)

- receive properties as inputs from so called “smart“ / “container“ parent components.

- are easily reusable.

Presentational Modules

LeanIX

```
@NgModule({
  declarations: [
    BookmarksListComponent,
    BookmarksListItemComponent,
    BookmarksActiveBookmarkInfoComponent
  ],
  imports: [
    CommonModule,
    LxCoreUIModule,
    LxPopoverModule,
    TranslateModule.forChild()
  ],
  providers: [],
  exports: [
    BookmarksListComponent,
    BookmarksListItemComponent,
    BookmarksActiveBookmarkInfoComponent
  ]
})
export class LxBookmarkUiModule { }
```

- No services
- CoreUI module consists of components that everyone needs, e.g. a loading spinner.
- <Feature>UI module consists of components, that are used for a specific feature.

Mock Services with Injection Tokens in Angular

LeanIX

```ts
/* ... */
import { InjectionToken } from '@angular/core';

export const BOOKMARK_READER = new InjectionToken<BookmarkReader>('BOOKMARK_READER');

export interface BookmarkReader {
  readBookmark(id: string): Observable<Bookmark>;
}
```

Injection Token definition

```ts
@Injectable()
export class StorybookBookmarkReaderService implements BookmarkReader {
  public readBookmark(id: string) {
    return of(FULL_BOOKMARKS.find(bookmark => bookmark.id === id));
  }
}
```

“Mock“ BookmarkReader for Storybook

```ts
@Component({
  changeDetection: ChangeDetectionStrategy.OnPush,
  /* ... */
})
export class BookmarksListItemComponent {
  /* ... */
  constructor(
    @Inject(DATE_FORMATS) private dateFormats: DateFormatsProvider,
    @Inject(BOOKMARK_READER) private bookmarkReader: BookmarkReader,
    /* ... */
  ) { }
  /* ... */
}
```

Component depends on BOOKMARK_READER implementation

```ts
storiesOf('Bookmarks|BookmarksList', module)
  .addDecorator(
    moduleMetadata({
      imports: [LxBookmarkUiModule],
      providers: [
        {
          provide: DATE_FORMATS,
          useClass: StorybookDateFormatsService
        },
        {
          provide: BOOKMARK_READER,
          useClass: StorybookBookmarkReaderService
        }
      ]
    })
  .add('list with different bookmarks', () => ({
    component: BookmarksListComponent,
    template: `
      <lx-bookmarks-list
        [activeBookmark]="activeBookmark"
        [bookmarkSuggestions]="bookmarkSuggestions">
      </lx-bookmarks-list>
    `,
    props: {
      activeBookmark: BookmarkHelpers.getBookmark(),
      bookmarkSuggestions: [/* ... */]
    }
  }));
```

Provide Storybook implementation in Angular module

LeanIX

Drawbacks?

Goal: Design System and Component Library

LeanIX

- We are using Zeroheight and Storybook to establish a common design language across all designers and frontend developers.
- A set of components, that can be reused in different applications and teams within LeanIX.
- Visual consistency 🌈

Resources

LeanIX

- https://github.com/storybookjs
- https://storybook.js.org/docs/basics/introduction/

Examples

- https://jetbrains.github.io/ring-ui/master/
- https://storybook-design-system.netlify.com/
- https://airbnb.io/react-dates/

Questions?

Great Place To Work. 2019 Beste Arbeitgeber™ ITK

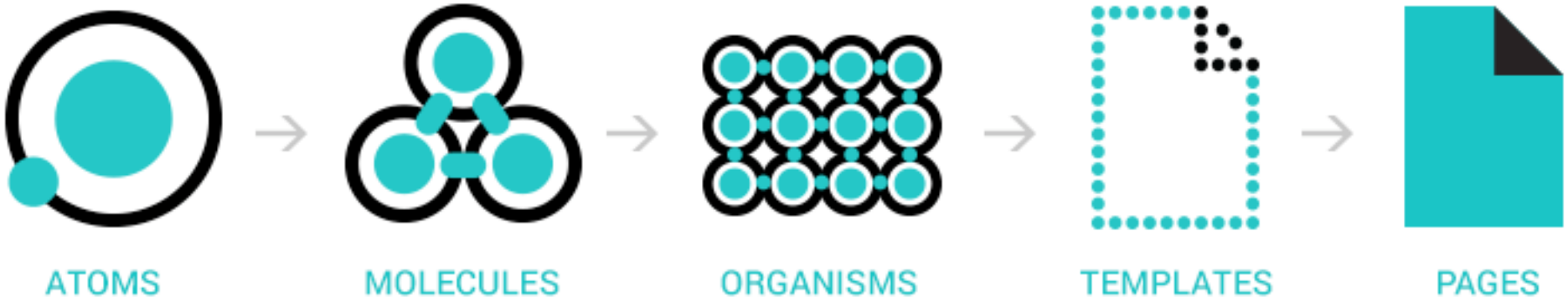
Extra Slides!?

LeanIX

"Storybook is a powerful frontend workshop environment tool that allows teams to design, build, and organize UI components (and even full screens!) without getting tripped up over business logic and plumbing."

Brad Frost
Author of Atomic Design

Atomic Design



Quelle: https://trespuntoscomunicacion.es/blog-diseno-web/atomic-design-ultima-tendencia-en-diseno-web/